Statistical Science
2016, Vol. 31, No. 1, 40–60
DOI: 10.1214/15-STS531

Analysis Methods for Computer Experiments: How to Assess and What Counts?

Hao Chen, Jason L. Loeppky, Jerome Sacks and William J. Welch

Abstract. Statistical methods based on a regression model plus a zero-mean Gaussian process (GP) have been widely used for predicting the output of a deterministic computer code. There are many suggestions in the literature for how to choose the regression component and how to model the correlation structure of the GP. This article argues that comprehensive, evidence-based assessment strategies are needed when comparing such modeling options. Otherwise, one is easily misled. Applying the strategies to several computer codes shows that a regression model more complex than a constant mean either has little impact on prediction accuracy or is an impediment. The choice of correlation function has modest effect, but there is little to separate two common choices, the power exponential and the Matérn, if the latter is optimized with respect to its smoothness. The applications presented here also provide no evidence that a composite of GPs provides practical improvement in prediction accuracy. A limited comparison of Bayesian and empirical Bayes methods is similarly inconclusive. In contrast, we find that the effect of experimental design is surprisingly large, even for designs of the same type with the same theoretical properties.

Key words and phrases: Correlation function, Gaussian process, kriging, prediction accuracy, regression.

1. INTRODUCTION

Over the past quarter century a literature beginning with Sacks, Schiller and Welch (1989), Sacks et al. (1989, in this journal), Currin et al. (1991), and O'Hagan (1992) has grown to explore the design and analysis of computer experiments. Such an experiment is a designed set of runs of a mathematical model implemented as a computer code. Running the code with vector-valued input $\mathbf{x}$ gives the output value, $y(\mathbf{x})$, assumed real-valued and deterministic: Running the code again with the same value for $\mathbf{x}$ does not change $y(\mathbf{x})$. A design D is a set of n runs at n configurations of $\mathbf{x}$, and an objective of primary interest is to use the data (inputs and outputs) to predict, via a statistical model, the output of the code at untried input values.

The basic approach to the statistical model typically adopted starts by thinking of the output function, $y(\mathbf{x})$, as being in a class of functions with prior distribution a Gaussian process (GP). The process has mean μ, which may be a regression function in $\mathbf{x}$, and a covariance function, $\sigma^2 R$, from a specified parametric family. Prediction is then made through the posterior mean given the data from the computer experiment, with some variations depending on whether a maximum likelihood (empirical Bayes) or fuller Bayesian implementation is used. While we partially address those variations in this article, our main thrusts are the practical questions faced by the user: What regression

Hao Chen is a Ph.D. candidate and William J. Welch is Professor, Department of Statistics, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: hao.chen@stat.ubc.ca; will@stat.ubc.ca). Jason L. Loeppky is Associate Professor, Statistics, University of British Columbia, Kelowna, BC V1V 1V7, Canada (e-mail: jason.loeppky@ubc.ca). Jerome Sacks is Director Emeritus, National Institute of Statistical Sciences, Research Triangle Park, North Carolina 27709, USA (e-mail: sacks@niss.org).

function and correlation function should be used? Does it matter?

We will call a GP model with specified regression and correlation functions a Gaussian stochastic process (GaSP) model. For example, GaSP(Const, PowerExp) will denote a constant (intercept only) regression and the power-exponential correlation function. The various regression models and correlation functions under consideration in this article will be defined in Section 2.

The rationale for the GaSP approach stems from the common situation that the dimension, d, of the space of inputs is not small, the function is fairly complex to model, and n is not large (code runs are expensive), hindering the effectiveness of standard methods (e.g., polynomials, splines, MARS) for producing predictions. The GaSP approach allows for a flexible choice of approximating models that adapts to the data and, more tellingly, has proved effective in coping with complex codes with moderately high d and scarce data. There is a vast literature treating an analysis in this context.

This article studies the impact on prediction accuracy of the particular model specifications commonly used, particularly μ, R, n, D. The goals are twofold. First, we propose a more evidence-based approach to distinguish what may be important from the unimportant and what may need further exploration. Second, our application of this approach to various examples leads to some specific recommendations.

Assessing statistical strategies for the analysis of a computer experiment often mimics what is done for physical experiments: a method is proposed, applied in examples—usually few in number—and compared to other methods. Where possible, formal, mathematical comparisons are made; otherwise, criteria for assessing performance are empirical. An initial empirical study for a physical experiment is forced to rely on the specific data of that experiment and, while different analysis methods may be applied, all are bound by the single data set. There are limited opportunities to vary sample size or design.

Computer experiments provide richer opportunities. Fast-to-run codes enable a laboratory to investigate the relative merits of an analysis method. A whole spectrum of "replicate" experiments can be conducted for a single code, going beyond a thimbleful of "anecdotal" reports.

The danger of being misled by anecdotes can be seen in the following example. The borehole function [Morris, Mitchell and Ylvisaker, 1993, also given in the supplementary material (Chen et al., 2016)] is frequently used to illustrate methodology for computer experiments. A 27-run orthogonal array (OA) in the 8 input factors was proposed as a design, following Joseph, Hung and Sudjianto (2008). The 27 runs were analyzed via GaSP with a specific R (the Gaussian correlation function described in Section 2) and with two choices for μ: a simple constant (intercept) versus a method to select linear terms (SL), also described in Section 2. The details of these alternative models are not important for now, just that we are comparing two modeling methods. A set of 10,000 test points selected at random in the 8-dimensional input space was then predicted. The resulting values of the root mean squared error (RMSE) measure defined in (2.5) of Section 2 were 0.141 and 0.080 for the constant and SL regression models, respectively.

With the SL approach reducing the RMSE by about 43% relative to a model with a constant mean, does this example provide powerful evidence for using regression terms in the GaSP model? Not quite. We replicated the experiment with the same choices of μ, R, n and the same test-data, but the training data came from a theoretically equivalent 27-run OA design. (There are many equivalent OAs, e.g., by permuting the labels between columns of a fixed OA.) The RMSE values in the second analysis were 0.073 and 0.465 for the constant and SL models respectively; SL produced about 6 times the error relative to a constant mean—the evidence *against* using regression terms is even more powerful!

A broader approach is needed. The one we take is laid out starting in Section 2, where we specify the alternatives considered for the statistical model's regression component and correlation function, and define the assessment measures to be used. We focus on the fundamental criterion of prediction accuracy (uncertainty assessment is discussed briefly in Section 6.1). In Section 3 we outline the basic idea of generating repeated data sets for any given example. The method is (exhaustively) implemented for several fast codes, including the aforementioned borehole function, along with some choices of n and D. In Section 4 the method is adapted to slower codes where data from only one experiment are available. Ideally, the universe of computer experiments is represented by a set of test problems and assessment criteria, as in numerical optimization (Dixon and Szegö, 1978); the codes and data sets investigated in this article and its supplementary material (Chen et al., 2016) are but an initial set. In Section 5 other modeling strategies are compared. Finally,

in Sections 6 and 7 we make some summarizing comments, conclusions and recommendations.

The article's main findings are that regression terms are unnecessary or even sometimes an impediment, the choice of R matters for less smooth functions, and that the variability of performance of a method for the same problem over equivalent designs is alarmingly high. Such variation can mask the differences in analysis methods, rendering them unimportant and reinforcing the message that light evidence leads to flimsy conclusions.

2. STATISTICAL MODELS, EXPERIMENTAL DESIGN, AND ASSESSMENT

A computer code output is denoted by $y(\mathbf{x})$ where the input vector, $\mathbf{x} = (x_1, \ldots, x_d)$, is in the d-dimensional unit cube. As long as the input space is rectangular, transforming to the unit cube is straightforward and does not lose generality. Suppose n runs of the code are made according to a design D of input vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$ in $[0,1]^d$, resulting in data $\mathbf{y} = (y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(n)}))^T$. The goal is to predict $y(\mathbf{x})$ at untried $\mathbf{x}$.

The GaSP approach uses a regression model and GP prior on the class of possible $y(\mathbf{x})$. Specifically, $y(\mathbf{x})$ is a priori considered to be a realization of

$$Y(\mathbf{x}) = \mu(\mathbf{x}) + Z(\mathbf{x}), \tag{2.1}$$

where $\mu(\mathbf{x})$ is the regression component, the mean of the process, and $Z(\mathbf{x})$ has mean 0, variance σ^2, and correlation function R.

2.1 Choices for the Correlation Function

Let $\mathbf{x}$ and $\mathbf{x}'$ denote two values of the input vector. The correlation between $Z(\mathbf{x})$ and $Z(\mathbf{x}')$ is denoted by $R(\mathbf{x}, \mathbf{x}')$. Following common practice, $R(\mathbf{x}, \mathbf{x}')$ is taken to be a product of 1-d correlation functions in the distances $h_j = |x_j - x_j'|$, that is, $R(\mathbf{x}, \mathbf{x}') = \prod_{j=1}^d R_j(h_j)$. We mainly consider four choices for R_j:

- Power exponential (abbreviated PowerExp):

$$R_j(h_j) = \exp\bigl(-\theta_j h_j^{p_j}\bigr), \tag{2.2}$$

  with $\theta_j \geq 0$ and $1 \leq p_j \leq 2$ controlling the sensitivity and smoothness, respectively, of predictions of y with respect to x_j.
- Squared exponential or Gaussian (abbreviated Gauss): the special case of PowerExp in (2.2) with all $p_j = 2$.
- Matérn:

$$R_j(h_j) = \frac{1}{\Gamma(\nu_j)2^{(\nu_j-1)}}(\theta_j h_j)^{\nu_j} K_{\nu_j}(\theta_j h_j), \tag{2.3}$$

  where Γ is the Gamma function, and K_{ν_j} is the modified Bessel function of order ν_j. Again, $\theta_j \geq 0$ is a sensitivity parameter. The Matérn class was recommended by Stein (1999), Section 2.7, for its control via $\nu_j > 0$ of the differentiability of the correlation function with respect to x_j, and hence that of the prediction function. With $\nu_j = 1 + \frac{1}{2}$ or $\nu_j = 2 + \frac{1}{2}$, there are 1 or 2 derivatives, respectively. We call these subfamilies Matérn-1 and Matérn-2. Similarly, we use Matérn-0 and Matérn-∞ to refer to the cases $\nu_j = 0 + \frac{1}{2}$ and $\nu_j \to \infty$. They give the exponential family [$p_j = 1$ in (2.2)], with no derivatives, and Gauss, which is infinitely differentiable. Matérn-0, 1, 2 are closely related to linear, quadratic, and cubic splines. We believe that little would be gained by incorporating smoother kernels (but less smooth than the analytic Matérn-∞) in the study.

  Consideration of the entire Matérn class for $\nu_j > 0$ is computationally cumbersome for the large numbers of experiments we will evaluate. Hence, what we call Matérn has ν_j optimized over the Matérn-0, Matérn-1, Matérn-2, and Matérn-∞ special cases, separately for each coordinate.
- Matérn-2: Some authors (e.g., Picheny et al., 2013) fix ν_j in the Matérn correlation function to give some differentiability. The Matérn-2 subfamily sets $\nu_j = 2 + \frac{1}{2}$ for all j, giving 2 derivatives.

More recently, other types of covariance functions have been recommended to cope with "apparently nonstationary" functions (e.g., Ba and Joseph, 2012). In Section 5.2 we will discuss the implications and characteristics of these options.

Given a choice for R_j and hence R, we define the $n \times n$ matrix $\mathbf{R}$ with element i, j given by $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ and the $n \times 1$ vector $\mathbf{r}(\mathbf{x}) = (R(\mathbf{x}, \mathbf{x}^{(1)}), \ldots, R(\mathbf{x}, \mathbf{x}^{(n)}))^T$ for any $\mathbf{x}$ where we want to predict.

2.2 Choices for the Regression Component

We explore three main choices for μ:

- Constant (abbreviated Const): $\mu(\mathbf{x}) = \beta_0$.
- Full linear (FL): $\mu(\mathbf{x}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d$, that is, a full linear model in all input variables.
- Select linear (SL): $\mu(\mathbf{x})$ is linear in the x_j like FL but only includes selected terms.

The proposed algorithm for SL is as follows. For a given correlation family construct a default predictor with Const for μ. Decompose the predictive function (Schonlau and Welch, 2006) and identify all main effects that contribute more than $100/d$ percent to the total variation. These become the selected coordinates. Typically, large main effects have clear linear components. If a large effect lacks a linear component, little is lost by including a linear term. Inclusion of possible nonlinear trends can be pursued at considerable computational cost; we do not routinely do so, but in Section 4.1 we do include a regression model with nonlinear terms in x_j.

All candidate regression models considered can be written in the form

$$\mu(\mathbf{x}) = \beta_0 + \beta_1 f_1(\mathbf{x}) + \cdots + \beta_k f_k(\mathbf{x}),$$

where the functions $f_j(\mathbf{x})$ are known. The maximum likelihood estimate (MLE) of $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_k)^T$ is the generalized least-squares estimate $\hat{\boldsymbol{\beta}} = (\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F})^{-1} \cdot \mathbf{F}^T\mathbf{R}^{-1}\mathbf{y}$, where the $n \times (k+1)$ matrix $\mathbf{F}$ has $(1, f_1(\mathbf{x}^{(i)}), \ldots, f_k(\mathbf{x}^{(i)}))$ in row i. This is also the Bayes posterior mean with a diffuse prior for $\boldsymbol{\beta}$.

Early work (Sacks, Schiller and Welch, 1989) suggested that there is little to be gained (and maybe even something to lose) by using other than a constant term for μ. In addition, Lim et al. (2002) showed that polynomials can be exactly predicted with a minimal number of points using the Gauss correlation function, provided one lets the $\theta_j \to 0$. These points underline the fact that a GP prior can capture the complexity in the output of the code, suggesting that deploying regression terms is superfluous. The evidence we report later supports this impression.

2.3 Prediction

Predictions are made as follows. For given data and values of the parameters in R, the mean of the posterior predictive distribution of $y(\mathbf{x})$ is

$$\hat{y}(\mathbf{x}) = \hat{\mu}(\mathbf{x}) + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}}), \tag{2.4}$$

where $\hat{\mu}(\mathbf{x}) = \hat{\beta}_0 + \hat{\beta}_1 f_1(\mathbf{x}) + \cdots + \hat{\beta}_k f_k(\mathbf{x})$.

In practice, the other parameters, σ^2 and those in the correlation function R of equations (2.2) or (2.3), must be estimated too. Empirical Bayes replaces all of $\boldsymbol{\beta}$, σ^2, and the correlation parameters in R by their MLEs (Welch et al., 1992). Various other Bayes-based procedures are available, including one fully Bayesian strategy described in Section 5.1. Our focus, however, is not on the particular Bayes-based methods employed but rather on assumptions about the form of the underlying GaSP model.

2.4 Design

For fast codes we typically use as a base design D an approximate maximin Latin hypercube design (mLHD, Morris and Mitchell, 1995), with improved low-dimensional space-filling properties (Welch et al., 1996). A few other choices, such as orthogonal arrays, are also investigated in Section 3.5, with a more comprehensive comparison of different classes of design the subject of another ongoing study. In any event, we show that even for a fixed class of design and fixed n there is substantial variation in prediction accuracy over equivalent designs. Conclusions based on a single design choice can be misleading.

The effect of n on prediction accuracy was explored by Sacks, Schiller and Welch (1989) and more recently by Loeppky, Sacks and Welch (2009); its role in the comparison of competing alternatives for μ and R will also be addressed in Section 3.5.

2.5 Measures of Prediction Error

In order to compare various forms of the predictor in (2.4) built from the n code runs, $\mathbf{y} = (y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(n)}))^T$, we need to set some standards. The gold standard is to assess the magnitude of prediction error via holdout (test) data, that is, in predicting N further runs, $y(\mathbf{x}_{\text{ho}}^{(1)}), \ldots, y(\mathbf{x}_{\text{ho}}^{(N)})$. The prediction errors are $\hat{y}(\mathbf{x}_{\text{ho}}^{(i)}) - y(\mathbf{x}_{\text{ho}}^{(i)})$ for $i = 1, \ldots, N$.

The performance measure we use is a normalized RMSE of prediction over the holdout data, denoted by $e_{\text{rmse,ho}}$. The normalization is the RMSE using the (trivial) predictor $\bar{y}$, the mean of the data from the runs in the experimental design, the "training" set. Thus,

$$e_{\text{rmse,ho}} = \frac{\sqrt{(1/N)\sum_{i=1}^{N}(\hat{y}(\mathbf{x}_{\text{ho}}^{(i)}) - y(\mathbf{x}_{\text{ho}}^{(i)}))^2}}{\sqrt{(1/N)\sum_{i=1}^{N}(\bar{y} - y(\mathbf{x}_{\text{ho}}^{(i)}))^2}}. \tag{2.5}$$

The normalization in the denominator puts $e_{\text{rmse,ho}}$ roughly on $[0, 1]$ whatever the scale of y, with 1 indicating no better performance than $\bar{y}$. The criterion is related to R^2 in regression, but $e_{\text{rmse,ho}}$ measures performance for a new test set and smaller values are desirable.

Similarly, worst-case performance can be defined as the normalized maximum absolute error. Results for this metric are reported in the supplementary material (Chen et al., 2016); the conclusions are the same. Other definitions (such as median absolute error) and other normalizations (such as those of Loeppky, Sacks and Welch, 2009) can be used, but without substantive effect on comparisons.

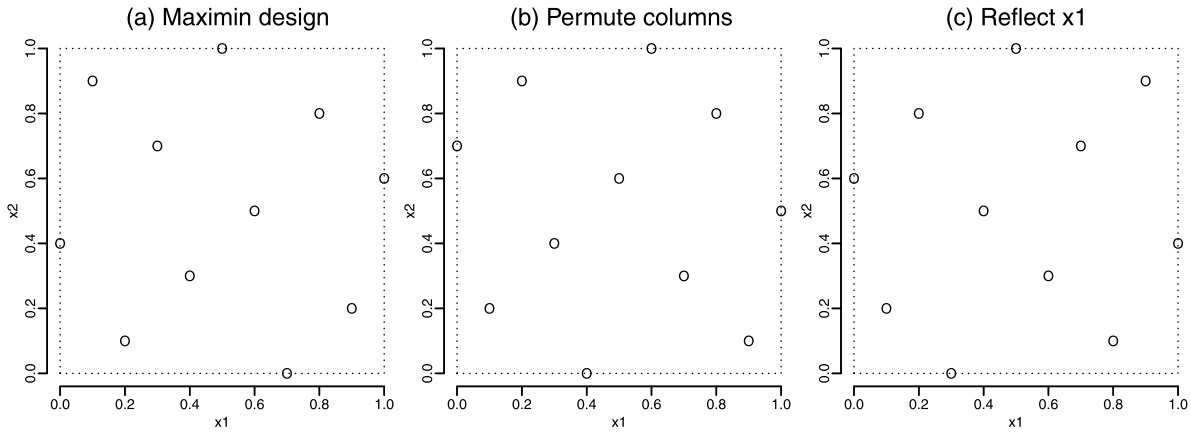

FIG. 1. *Equivalent designs for* $d = 2$ *and* $n = 11$: (a) *a base mLHD design*; (b) *the base design with labels permuted between columns*; *and* (c) *the base design with values in the* x_1 *column reflected around* $x_1 = 0.5$.

What are tolerable levels of error? Clearly, these are application-specific so that tighter thresholds would be demanded, say, for optimization than for sensitivity analysis. For general purposes we take the rule of thumb that $e_{\text{rmse,ho}} < 0.10$ is useful. For normalized maximum error it is plausible that the threshold could be much larger, say 0.25 or 0.30. These speculations are consequences of the experiences we document later, and are surely not the last word. The value of having thresholds is to provide benchmarks that enable assessing when differences among different methods or strategies are practically insignificant versus statistically significant.

3. FAST CODES

3.1 Generating a Reference Set for Comparisons

For fast codes under our control, large holdout sets can be obtained. Hence, in this section performance is measured through the use of a holdout (test) set of 10,000 points, selected as a random Latin hypercube design (LHD) on the input space.

With a fast code many designs and hence training data sets can easily be generated. We generate many equivalent designs by exploiting symmetries. For a simple illustration, Figure 1(a) shows a base mLHD for $d = 2$ and $n = 11$. Permuting the labels between columns of the design, that is, interchanging the x_1 and x_2 values as in Figure 1(b), does not change the interpoint distances or properties based on them such as the minimum distance used to construct mLHD designs. Similarly, reflecting the values within, say, the x_1 column around $x_1 = 0.5$ as in Figure 1(c), does not change the properties. In this sense the designs are equivalent.

In general, for any base design with good properties, there are $d!2^d$ equivalent designs and hence equivalent sets of training data available from permuting all column labels and reflecting within columns for a subset of inputs. For the borehole code mentioned in Section 1 and investigated more fully in Section 3.2, we have found that permuting between columns gives more variation in prediction accuracy than reflecting within columns. Thus, in this article for nearly all examples we only permute between columns: for $d = 4$ all 24 possible permutations, and for $d \geq 5$ a random selection of 25 different permutations. The example of Section 5.2 with $d = 2$ is the one exception. Because $y(x_1, x_2)$ is symmetric in x_1 and x_2, permuting between columns does not change the training data and we reflect within x_1 and/or x_2 instead.

The designs, generated data sets, and replicate analyses then serve as the reference set for a particular problem and provide the grounds on which variability of performance can be assessed. Given the setup of Section 2, we want to assess the consequences of making a choice from the menu of three regression models and four correlation functions.

3.2 Borehole Code

The first setting we will look at is the borehole code (Morris, Mitchell and Ylvisaker, 1993) mentioned in Section 1 and described in the supplementary material (Chen et al., 2016). It has served as a test bed in many contexts (e.g., Joseph, Hung and Sudjianto, 2008). We consider three different designs for the experiment: a 27-run, 3-level orthogonal array (OA), the same design used by Joseph, Hung and Sudjianto (2008); a 27-run mLHD; and a 40-run mLHD.

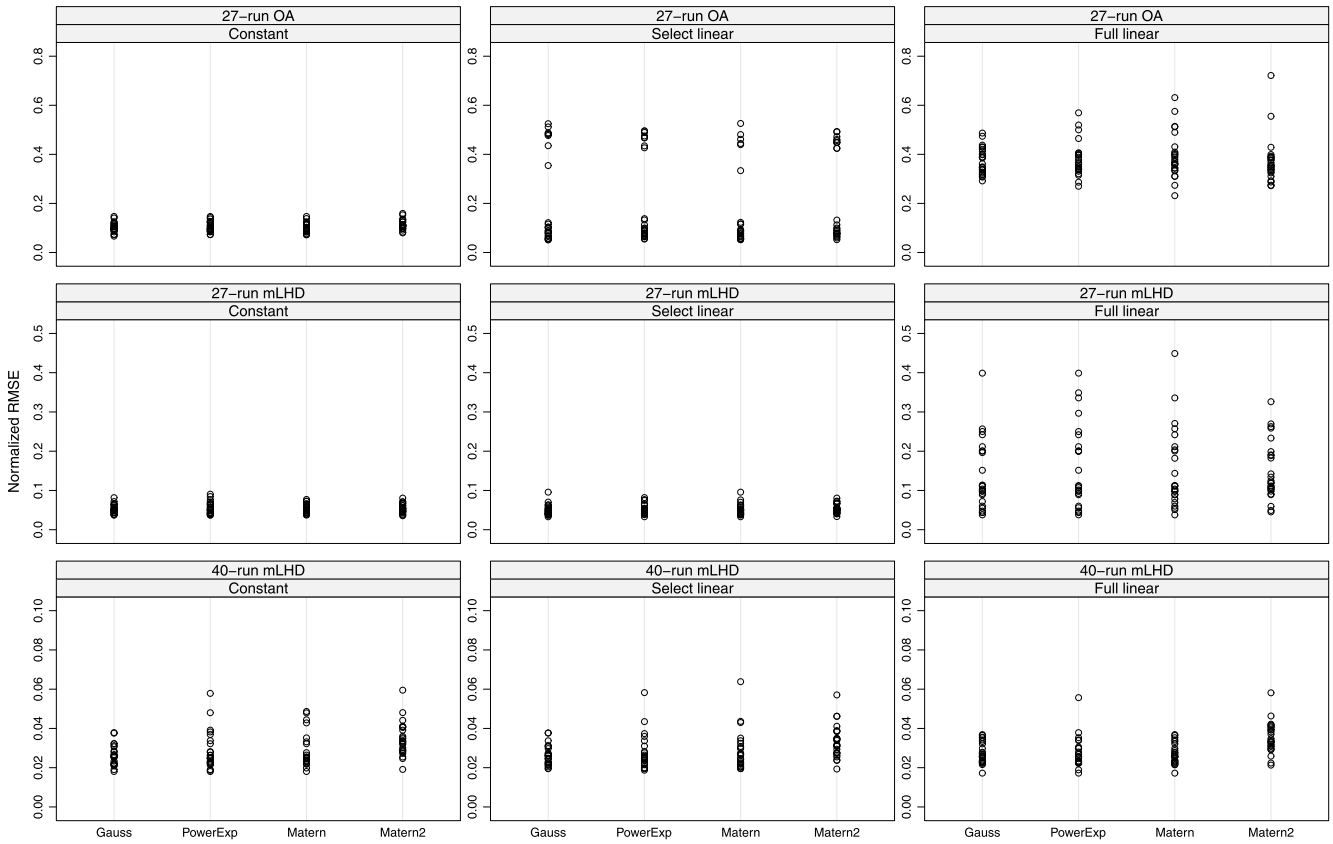

FIG. 2. *Borehole function*: *Normalized holdout RMSE of prediction*, $e_{\text{rmse,ho}}$, *for GaSP with all combinations of three regression models and four correlation functions*. *There are three base designs*: *a* 27*-run OA* (*top row*); *a* 27*-run mLHD* (*middle row*); *and a* 40*-run mLHD* (*bottom row*). *For each base design*, 25 *random permutations between columns give the* 25 *values of* $e_{\text{rmse,ho}}$ *in a dot plot*.

There are 12 possible modeling combinations from the four correlation functions and three regression models outlined in Sections 2.1 and 2.2. The SL choice for μ here is always the term x_1. Its main effect accounts for approximately 80% of the variation in predictions over the 8-dimensional input domain, and all analyses with a Const regression model choose x_1 and no other terms across all designs and all repeat experiments.

The top row of Figure 2 shows results with the 27-run OA design. For a given modeling strategy, 25 random permutations between columns of the 27-run OA lead to 25 repeat experiments (Section 3.1) and hence a reference set of 25 values of $e_{\text{rmse,ho}}$ shown as a dot plot. The results are striking. Relative to a constant regression model, the FL regression model has empirical distributions of $e_{\text{rmse,ho}}$ which are uniformly and substantially inferior, for all correlation functions. The SL regression also performs very poorly sometimes, but not always. To investigate the SL regression further, Figure 3 plots $e_{\text{rmse,ho}}$ for individual repeat experiments, comparing the GaSP(Const, Gauss) and GaSP(SL, Gauss) models. Consistent with the anecdotal comparisons in Section 1, the plot shows that

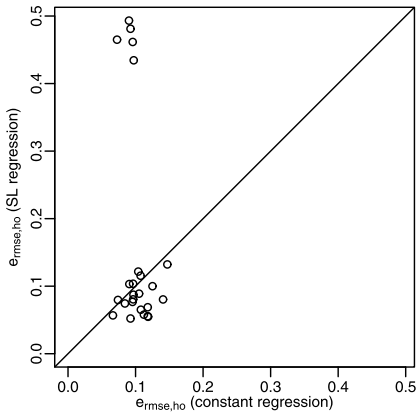

FIG. 3. *Borehole code*: *Normalized holdout RMSE of prediction*, $e_{\text{rmse,ho}}$, *for an SL regression model versus a constant regression model*. *The* 25 *points are from repeat experiments generated by* 25 *random permutations between columns of a* 27*-run OA*.

the SL regression model can give a smaller $e_{\text{rmse,ho}}$—this tends to happen when both methods perform fairly well—but the SL regression sometimes has very poor accuracy (almost 0.5 on the normalized RMSE scale). The top row of Figure 2 also shows that the choice of correlation function is far less important than the choice of regression model.

The results for the 27-run mLHD in the middle row of Figure 2 show that design can have a large effect on accuracy: every analysis model performs better for the 27-run mLHD than for the 27-run OA. (Note the vertical scale is different for each row of the figure.) The SL regression now performs about the same as the constant regression instead of occasionally much worse. There is no substantial difference in accuracy between the correlation functions. Indeed, the impact on accuracy of using the space-filling mLHD design instead of an OA is much more important than differences due to choice of the correlation function. The scaling in the middle row of plots somewhat mutes the considerable variation in accuracy still present over the 25 equivalent mLHD designs.

Increasing the number of runs to a 40-run mLHD (bottom row of Figure 2) makes a further substantial improvement in prediction accuracy. All 12 modeling strategies give $e_{\text{rmse,ho}}$ values of about 0.02–0.06 over the 25 repeat experiments. Although there is little systematic difference among strategies, the variation over equivalent designs is still striking in a relative sense.

The strikingly poor results from the SL regression model (sometimes) and the FL model (all 25 repeats) in the top row of Figure 2 may be explained as follows. The design is a 27-run OA with only 3 levels. In a simpler context, Welch et al. (1996) illustrated nonidentifiability of the important terms in a GaSP model when the design is not space-filling. The SL regression and, even more so, the FL regression complicate an already flexible GP model. The difficulty in identifying the important terms is underscored by the fact that for all 25 repeat experiments from the base 27-run OA, a least squares fit of a simple linear regression model in x_1 (with no other terms) gives $e_{\text{rmse,ho}}$ values close to 0.45. In other words, performance of GaSP(SL, Gauss) is sometimes similar to fitting just the important x_1 linear trend. The performance of GaSP(FL, Gauss) is highly variable and sometimes even worse than simple linear regression.

Welch et al. (1996) argued that model identifiability is, not surprisingly, connected with confounding in the design. The confounding in the base 27-run OA is complex. While it is preserved in an overall sense by permutations between columns, how the confounding structure aligns with the important inputs among $x_1, \ldots, x_8$ will change across the 25 repeat experiments. Hence, the impact of confounding on nonidentifiability will vary.

In contrast, accuracy for the space-filling design in the middle row of Figure 2 is much better, even with only 27 runs. The SL regression model performs as accurately as the Const model (but no better); only the even more complex FL regression runs into difficulties. Again, this parallels the simpler Welch et al. (1996) example, where model identification was less problematic with a space-filling design and largely eliminated by increasing the sample size (the bottom row of Figure 2).

3.3 G-Protein Code

A second application, the G-protein code used by Loeppky, Sacks and Welch (2009) and described in the supplementary material (Chen et al., 2016), consists of a system of ODE's with 4-dimensional input.

Figure 4 shows $e_{\text{rmse,ho}}$ for the three regression models (here SL selects x_2, x_3 as inputs with large effects) and four correlation functions. The results in the top row are for a 40-run mLHD. With $d = 4$, all 24 possible permutations between columns of a single base design lead to 24 data sets and hence 24 $e_{\text{rmse,ho}}$ values. The dot plots in the top row have similar distributions across the 12 modeling strategies. As these empirical distributions have most $e_{\text{rmse,ho}}$ values above 0.1, we try increasing the sample size with an 80-run mLHD. This has a substantial effect on accuracy, with all modeling methods giving $e_{\text{rmse,ho}}$ values of about 0.06 or less.

Thus, for the G-protein application, none of the three choices for μ or the four choices for R matter. The variation among equivalent designs is alarmingly large in a relative sense, dwarfing any impact of modeling strategy.

3.4 PTW Code

Results for a third fast-to-run code, PTW (Preston, Tonks and Wallace, 2003), are in the supplementary material (Chen et al., 2016). It has 11 inputs. We took a mLHD with $n = 110$ as the base design for the reference set. Prior information from engineers suggested incorporating linear x_1 and x_2 terms; SL also included x_3. No essential differences among μ or R emerged, but again there is a wide variation over equivalent designs.

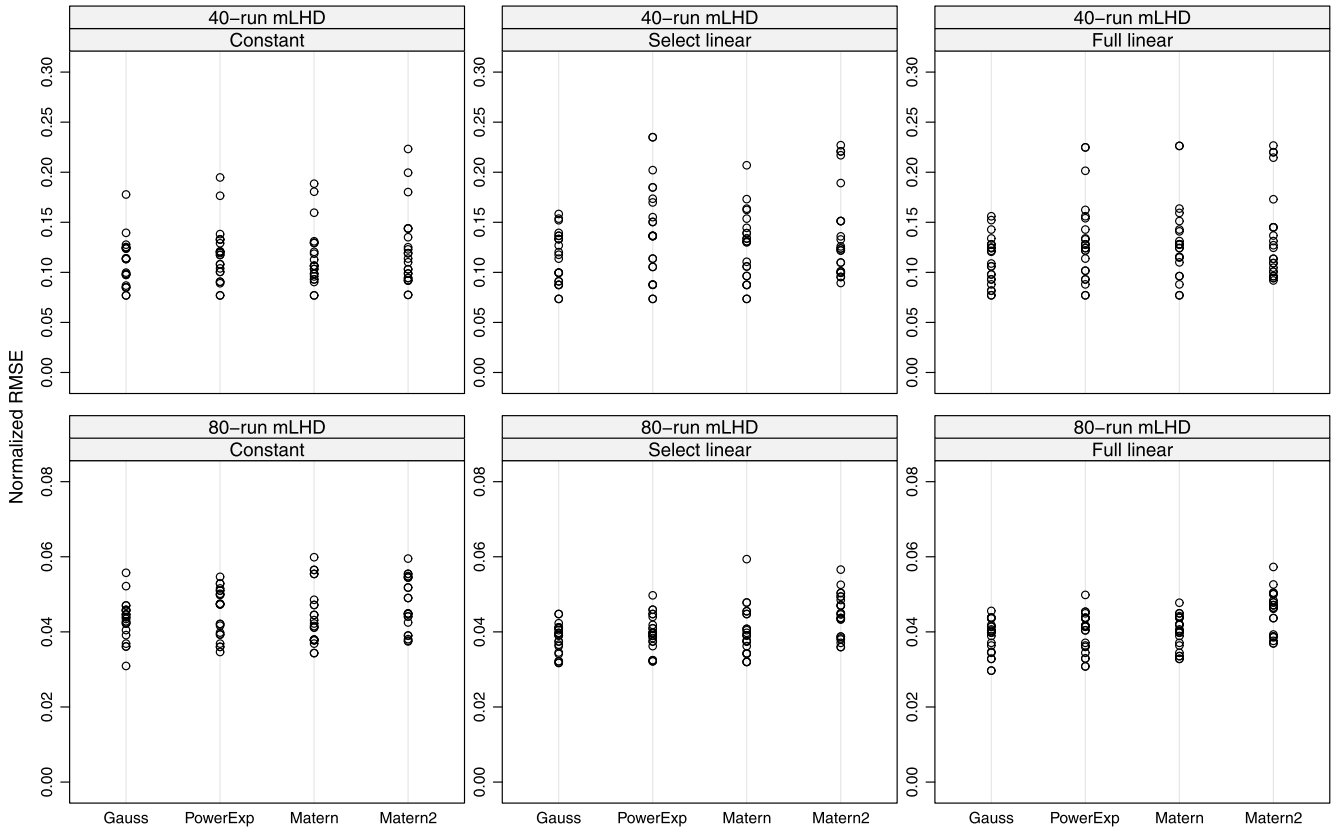

FIG. 4. *G-protein code*: *Normalized holdout RMSE of prediction*, $e_{\mathrm{rmse,ho}}$, *for all combinations of three regression models and four correlation functions*. *There are two base designs*: *a* 40*-run mLHD* (*top row*); *and an* 80*-run mLHD* (*bottom row*). *For each base design*, *all* 24 *permutations between columns give the* 24 *values of* $e_{\mathrm{rmse,ho}}$ *in each dot plot*.

3.5 Effect of Design

The results above document a significant, seldom recognized role of design: different, even equivalent, designs can have a greater effect on performance than the choice of μ, R. Moreover, without prior information, there is no way to assure that the choice of design will be one of the good ones in the equivalence class. Whether sequential experimentation, if feasible, can produce a more advantageous solution needs exploring.

The contrast between the results for borehole 27-run OA and the 27-run mLHD is a reminder of the importance of using designs that are space-filling, a quality widely appreciated. It is no secret that the choice of sample size, n, has a strong effect on performance as evidenced in the results for the 40-point mLHD contrasted with those for the 27-point mLHD. A more penetrating study of the effect of n was conducted by Loeppky, Sacks and Welch (2009). That FL does as well as Const and SL for the Borehole 40-point mLHD but performs badly for either of the two 27-point designs, and that none of the regression choices matter for the G-protein 40-point design or for the PTW 110-point design, suggests that "everything" works if n is large enough.

In summary, the choice of n and the choice of D given n can have huge effects. But have we enough evidence that choice of μ matters only in limited contexts (such as small n or poor design) and that choice of R does not matter? So far we have dealt with only a handful of simple, fast codes; it is time to consider more complex codes.

4. SLOW CODES

For complex costly-to-run codes, generating substantial holdout data or output from multiple designs is infeasible. Similarly, for codes where we only have reported data, new output data are unavailable. Forced to depend on what data are at hand leads us to rely on cross-validation methods for generating multiple designs and holdout sets, through which we can assess the effect of variability not solely in the designs but also,

and inseparably, in the holdout target data. We know from Section 3 that variability due to designs is considerable, and it is no surprise that variability in holdout sets would lead to variability in predictive performance. The utility then of the created multiple designs and holdout sets is to compare the behavior of different modeling choices under varying conditions rather than relying on a single quantity attached to the original, unique data set.

Our approach is simply to delete a subset from the full data set, use the remaining data to produce a predictor, and calculate the (normalized) RMSE from predicting the output in the deleted (holdout) subset. Repeating this for a number (25 is what we use) of subsets gives some measure of variability and accuracy. In effect, we create 25 designs and corresponding holdout sets from a single data set and compare consequences arising from different choices for predictors.

The details described in the applications below differ somewhat depending on the particular application. In the example of Section 4.1—a reflectance model for a plant canopy—there are, in fact, limited holdout data but no data from multiple designs. In the volcano-eruption example of Section 4.2 and the sea-ice model of Section 4.3 holdout data are unavailable.

4.1 Nilson–Kuusk Model

An ecological code modeling reflectance for a plant canopy developed by Nilson and Kuusk (1989) was used by Bastos and O'Hagan (2009) to illustrate diagnostics for GaSP models. With 5-dimensional input, two computer experiments were performed: the first using a 150-run random LHD and the second with an independently chosen LHD of 100 points.

We carry out three studies based on the same data. The first treats the 100-point LHD as the experiment and the 150-point set as a holdout sample. The second study reverses the roles of the two LHDs. A third study, extending one done by Bastos and O'Hagan (2009), takes the 150-run LHD, augments it with a random sample of 50 points from the 100-point LHD, takes the resulting 200-point subset as the experimental design for training the statistical model, and uses the remaining $N = 50$ points from the 100-run LHD to form the holdout set in the calculation of $e_{\text{rmse,ho}}$. By repeating the sampling of the 50 points 25 times, we get 25 replicate experiments, each with the same base 150 runs but differing with respect to the additional 50 training points and the holdout set.

In addition to the linear regression choices we have studied so far, we also incorporate a regression model identified by Bastos and O'Hagan (2009): an intercept, linear terms in the inputs $x_1, \ldots, x_4$, and a quartic polynomial in x_5. We label this model "Quartic." All analyses are carried out with the output y on a log scale, based on standard diagnostics for GaSP models (Jones, Schonlau and Welch, 1998).

TABLE 1
Nilson–Kuusk model: Normalized holdout RMSE of prediction, $e_{\text{rmse,ho}}$, for four regression models and four correlation functions. The experimental data are from a 100-run LHD, and the holdout set is from a 150-run LHD

	$e_{\text{rmse,ho}}$			
	Correlation function			
Regression model	**Gauss**	**PowerExp**	**Matérn-2**	**Matérn**
Constant	0.116	0.099	0.106	0.102
Select linear	0.115	0.099	0.106	0.105
Full linear	0.110	0.099	0.104	0.104
Quartic	0.118	0.103	0.107	0.106

Table 1 summarizes the results of the study with the 100-point LHD as training data and the 150-point set as a holdout sample. It shows the choice for μ is immaterial: the constant mean is as good as any. For the correlation function, Gauss is inferior to the other choices, there is some evidence that Matérn is preferred to Matérn-2, and there is little difference between PowerExp and Matérn, the best performers. Similar results pertain when the 150-run LHD is used for training and the 100-run set for testing [Table 4 in the supplementary material (Chen et al., 2016)].

The dot plots in Figure 5 for the third study are even more striking in exhibiting the inferiority of $R =$ Gauss and the lack of advantages for any of the nonconstant regression functions. The large variability in performance among designs and holdout sets is similar to that seen for the fast-code replicate experiments of Section 3. The perturbations of the experiment, from random sampling here, appear to provide a useful reference set for studying the behavior of model choices.

The large differences in prediction accuracy among the correlation functions, not seen in Section 3, deserve some attention. An overly smooth correlation function—the Gaussian—does not perform as well as the Matérn and power-exponential functions here. The latter two have the flexibility to allow needed rougher realizations. With the 150-run design and the constant regression model, for instance, the maximum of the log likelihood increases by about 50 when the power exponential is used instead of the Gaussian, with four of the p_j in (2.2) taking values less than 2.

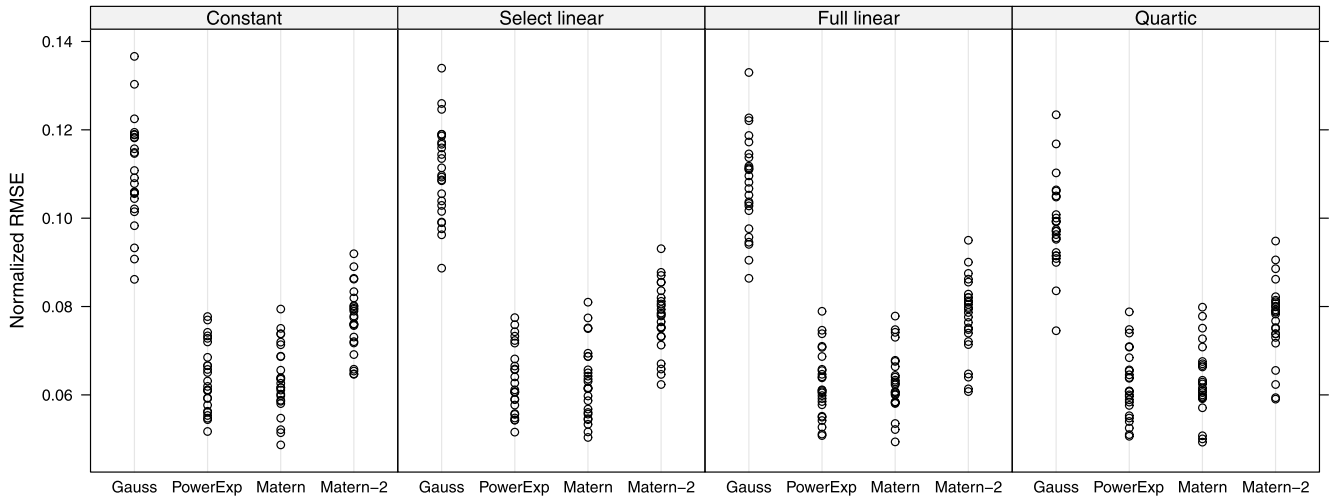

FIG. 5. *Nilson–Kuusk code*: *Normalized holdout RMSE of prediction*, $e_{\mathrm{rmse,ho}}$, *for four regression models and four correlation functions. Twenty-five designs are created from a* 150*-run LHD base plus* 50 *random points from a* 100*-run LHD. The remaining* 50 *points in the* 100*-run LHD form the holdout set for each repeat.*

The estimated main effect (Schonlau and Welch, 2006) of x_5 in Figure 6 from the GaSP(Const, PowerExp) model shows that x_5 has a complex effect. It is also a strong effect, accounting for about 90% of the total variance of the predicted output over the 5-dimensional input space. Bastos and O'Hagan (2009) correctly diagnosed the complexity of this trend. Modeling it via a quartic polynomial in x_5 has little impact on prediction accuracy, however. The correlation structure of the GaSP is able to capture the trend implicitly just as well.

4.2 Volcano Model

A computer model studied by Bayarri et al. (2009) models the process of pyroclastic flow (a fast-moving current of hot gas and rock) from a volcanic eruption.

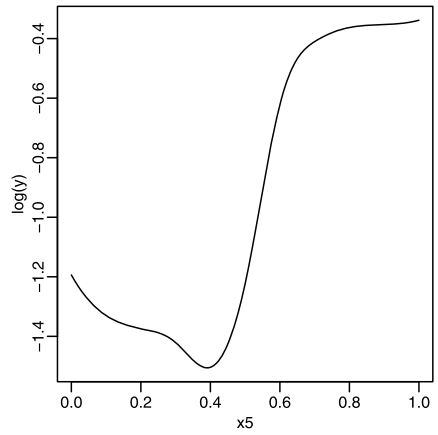

FIG. 6. *Nilson–Kuusk code*: *Estimated main effect of* x_5.

The inputs varied are as follows: initial volume, x_1, and direction, x_2, of the eruption. The output, y, is the maximum (over time) height of the flow at a location. A 32-run data set provided by Elaine Spiller [different from that reported by Bayarri et al. (2009) but a similar application] is available in the supplementary material (Chen et al., 2016). Plotting the data shows the output has a strong trend in x_1, and putting a linear term in the GaSP surrogate, as modeled by Bayarri et al. (2009), is natural. But is it necessary?

The nature of the data suggests a transformation of y could be useful. The one used by Bayarri et al. (2009) is $\log(y+1)$. Diagnostic plots (Jones, Schonlau and Welch, 1998) from using $\mu = \text{Const}$ and $R = \text{Gauss}$ show that the log transform is reasonable, but a square-root transformation is better still. We report analyses for both transformations.

The regression functions considered are Const, SL ($\beta_0 + \beta_1 x_1$), full linear, and quadratic ($\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2$), because the estimated effect of x_1 has a strong trend growing faster than linearly when looking at main effects from the surrogate obtained using $\sqrt{y}$ and GaSP(Const, PowerExp).

Analogous to the approach in Section 4.1, repeat experiments are generated by random sampling of 25 runs from the 32 available to comprise the design for model fitting. The remaining 7 runs form the holdout set. This is repeated 25 times, giving 25 $e_{\mathrm{rmse,ho}}$ values in the dot plots of Figure 7. The conclusions are much like those in Section 4.1: there is usually no need to go beyond $\mu = \text{Const}$ and PowerExp is preferred

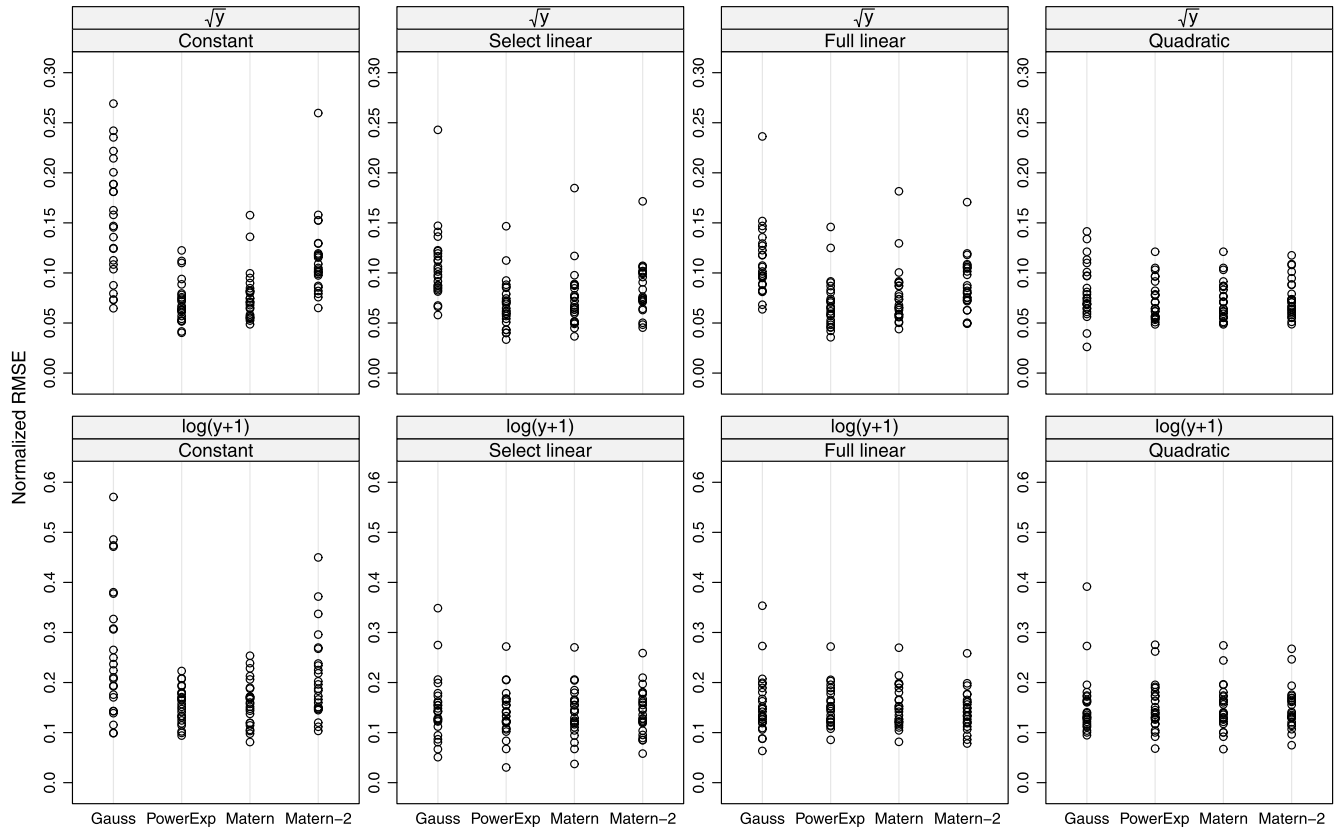

FIG. 7. *Volcano model*: *Normalized holdout RMSE*, $e_{\mathrm{rmse,ho}}$, *for four regression models and four correlation functions*. *The output variable is either* $\sqrt{y}$ *or* $\log(y+1)$.

to Gauss. The failure of Gauss in the two "slow" examples considered thus far is surprising in light of the widespread use of the Gauss correlation function.

4.3 Sea-Ice Model

The Arctic sea-ice model studied in Chapman et al. (1994) and in Loeppky, Sacks and Welch (2009) has 13 inputs, 4 outputs, and 157 available runs. The previous studies found modest prediction accuracy of GaSP(Const, PowerExp) surrogates for two of the outputs (ice mass and ice area) and poor accuracy for the other two (ice velocity and ice range). The question arises whether use of linear regression terms can increase accuracy to acceptable levels. Using a sampling process like that in Section 4.2 leads to the results in the supplementary material (Chen et al., 2016), where the answer is no: there is no help from $\mu = \mathrm{SL}$ or FL, nor from changing R. Indeed, FL makes accuracy much worse sometimes.

5. OTHER MODELING STRATEGIES

Clearly, we have not studied all possible paths to GaSP modeling that one might take in a computer experiment. In this section we address several others, some in more detail, and point to issues that could be addressed in the fashion described above.

5.1 Full Bayes

A number of full Bayes approaches have been employed in the literature. They go beyond the statistical formulation using a GP as a prior on the class of functions and assign prior distributions to all parameters, particularly those of the correlation function. For illustration, we examine the GEM-SA implementation of Kennedy (2004), which we call Bayes-GEM-SA. One key aspect is its reliance on $R = \mathrm{Gauss}$. It also uses the following independent prior distributions: $\beta_0 \propto 1$, $\sigma^2 \propto 1/\sigma^2$, and θ_j exponential with rate 0.01 (Kennedy, 2004). When comparing its predictive accuracy with GaSP, $\mu = \mathrm{Const}$ is used for all models.

For the borehole application, 25 repeat experiments are constructed for three designs, as in Section 3. The dot plots of $e_{\mathrm{rmse,ho}}$ in Figure 8 compare Bayes-GEM-SA with the Gauss and PowerExp methods in Section 3 based on MLEs of all parameters. (The method CGP

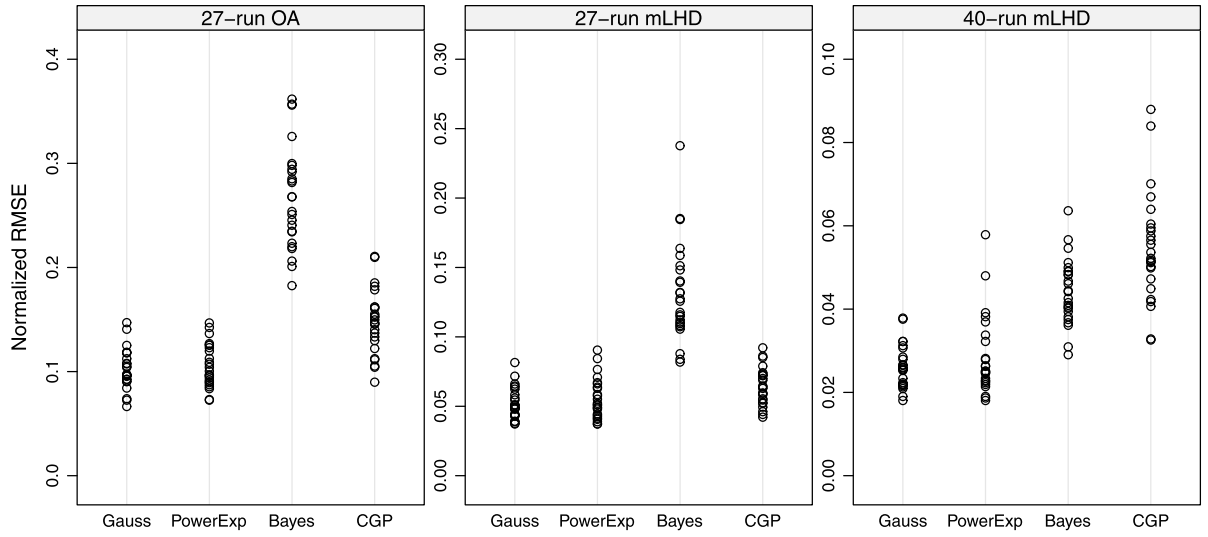

FIG. 8. *Borehole function*: *Normalized holdout RMSE of prediction*, $e_{\text{rmse,ho}}$, *for GaSP*(*Const*, *Gauss*), *GaSP*(*Const*, *PowerExp*), *Bayes-GEM-SA*, *and CGP. There are three base designs*: *a* 27*-run OA* (*left*), *a* 27*-run mLHD* (*middle*), *and a* 40*-run mLHD* (*right*). *For each base design*, 25 *random permutations between columns give the* 25 *values of* $e_{\text{rmse,ho}}$ *in a dot plot*.

and its dot plot are discussed in Section 5.2.) Bayes-GEM-SA is less accurate than either GaSP(Const, Gauss) or GaSP(Const, PowerExp).

Figure 9 similarly depicts results for the G-protein code. With the 40-run mLHD, the Bayesian and likelihood methods all perform about the same, giving only fair prediction accuracy. Increasing n to 80 improves accuracy considerably for all methods (the scales of the two plots are very different), far outweighing any systematic differences between their accuracies.

Bayes-GEM-SA performs as well as the GaSP methods for G-protein, not so well for Borehole with $n = 27$ but adequately for $n = 40$. Turning to the slow codes in Section 4, a different message emerges. Figure 10 for the Nilson–Kuusk model is based on 25 repeat designs constructed as for Figure 5 with a base design of 150 runs plus 50 randomly chosen from 100. The distributions of $e_{\text{rmse,ho}}$ for Bayes-GEM-SA and Gauss are similar, with PowerExp showing a clear advantage. Moreover, few of the Bayes $e_{\text{rmse,ho}}$ values meet the 0.10 threshold, while all the GaSP(Const, PowerExp) $e_{\text{rmse,ho}}$ values do. Bayes-GEM-SA uses the Gaussian correlation function, which performed relatively

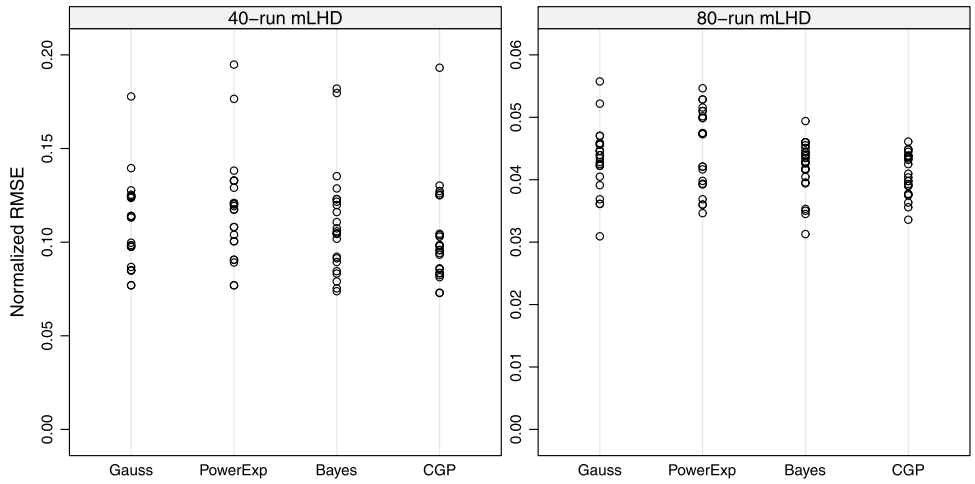

FIG. 9. *G-protein*: *Normalized holdout RMSE of prediction*, $e_{\text{rmse,ho}}$, *for GaSP*(*Const*, *Gauss*), *GaSP*(*Const*, *PowerExp*), *Bayes-GEM-SA*, *and CGP. There are two base designs*: *a* 40*-run mLHD* (*left*); *and an* 80*-run mLHD* (*right*). *For each base design*, *all* 24 *permutations between columns give the* 24 *values of* $e_{\text{rmse,ho}}$ *in a dot plot*.

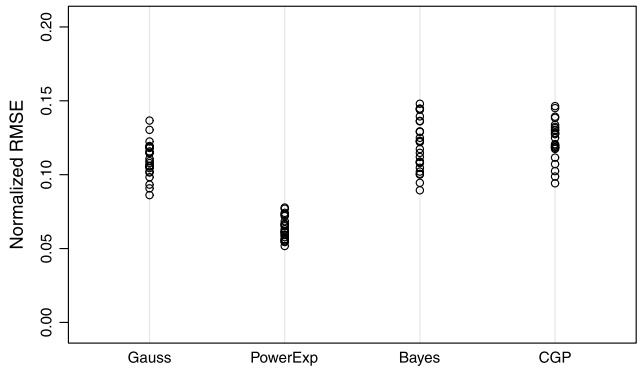

FIG. 10. *Nilson–Kuusk model*: *Normalized holdout RMSE of prediction*, $e_{\text{rmse,ho}}$, *for GaSP*(*Const*, *Gauss*), *GaSP*(*Const*, *PowerExp*), *Bayes-GEM-SA*, *and CGP.*

poorly in Section 4; the disadvantage carries over to the Bayesian method here.

The results in Figure 11 for the volcano code are for the 25 repeat experiments described in Section 4. Here again PowerExp dominates Bayes and for the same reasons as for the Nilson–Kuusk model. For the $\sqrt{y}$ transformation, all but a few GaSP(Const, PowerExp) $e_{\text{rmse,ho}}$ values meet the 0.10 threshold, in contrast to Bayes where all but a few do not.

These results are striking and suggest that Bayes methods relying on $R =$ Gauss need extension. The "hybrid" Bayes-MLE approach employed by Bayarri et al. (2009) estimates the correlation parameters in PowerExp by their MLEs, fixes them, and takes objective priors for μ and σ^2. The mean of the predictive distribution for a holdout output value gives the same prediction as GaSP(Const, PowerExp). Whether other "hybrid" forms can be brought to bear effectively needs exploration.

5.2 Nonstationarity

The use of stationary GPs as priors in the face of "nonstationary appearing" functions has attracted a measure of concern despite the fact that all functions with L_2-derivative can be approximated using PowerExp with enough data. Of course, there never are enough data. A relevant question is whether other priors, even stationary ones different from those in Section 2, are better reflective of conditions and lead to more accurate predictors.

West et al. (1995) employed a GP prior for $y(\mathbf{x})$ with two additive components: a smooth one for global trend and a rough one to model more local behavior. Recently, a similar "composite" GP (CGP) approach was advanced by Ba and Joseph (2012). These authors used two GPs, both with Gauss correlation. The first has correlation parameters θ_j in (2.2) constrained to be small for gradually varying longer-range trend, while the second has larger values of θ_j for shorter-range behavior. The second, local GP also has a variance that depends on $\mathbf{x}$, primarily as a way to cope with apparent nonstationary behavior. Does this composite approach offer an effective improvement to the simpler choices of Section 2?

We can apply CGP via its R library to the examples studied in Sections 3 and 4, much as was just done for Bayes-GEM-SA. The comparisons in Figure 8 for the borehole function show that GaSP and CGP have similar accuracy for the two 27-run designs. GaSP has smaller error than CGP for the 40-run mLHD,

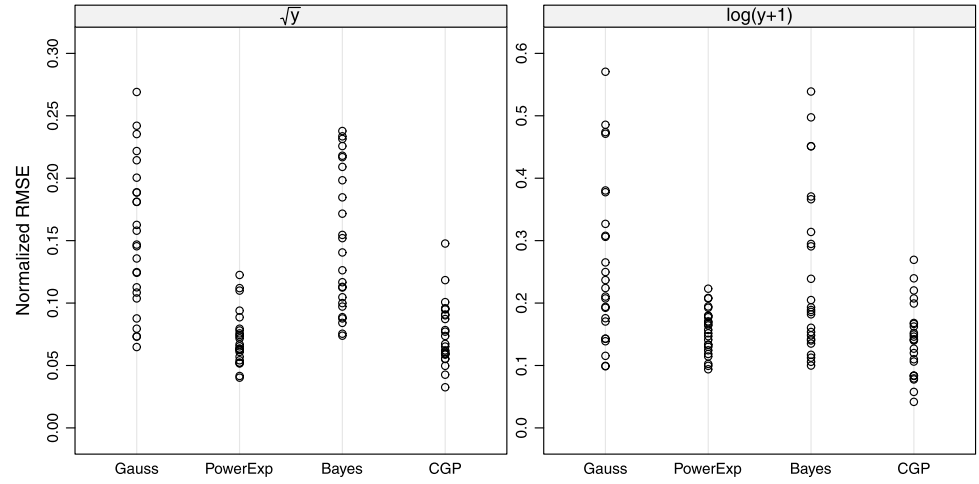

FIG. 11. *Volcano model*: *Normalized holdout RMSE of prediction*, $e_{\text{rmse,ho}}$, *for GaSP*(*Const*, *Gauss*), *GaSP*(*Const*, *PowerExp*), *Bayes-GEM-SA*, *and CGP.*

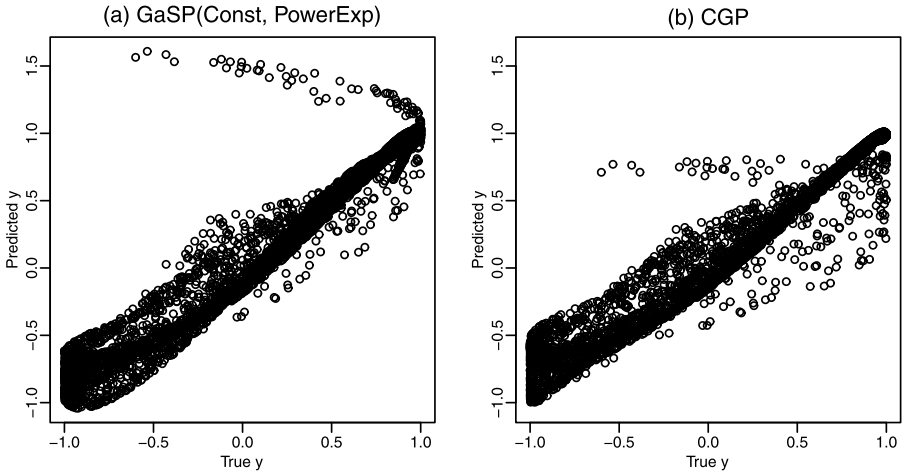

FIG. 12. 2-*d function*: *Holdout predictions versus true values of y from fitting* (a) *GaSP*(*Const*, *PowerExp*) *and* (b) *CGP.*

though both methods achieve acceptable accuracy. The results in Figure 9 for G-protein show little practical difference between any of the methods, including CGP. For these two fast-code examples, there is negligible difference between CGP and the GaSP methods. For the models of Section 4, however, conclusions are somewhat different. GaSP(Const, PowerExp) is clearly much more accurate than CGP for the Nilson–Kuusk model (Figure 10) and roughly equivalent for the volcano code (Figure 11).

Ba and Joseph (2012) gave several examples assessing the performance of CGP. For reasons noted in Sections 6.2 and 6.4 we only look at two.

10-*d example*. The test function is

$$y(\mathbf{x}) = -\sum_{j=1}^{10} \sin(x_j)\bigl(\sin\bigl(jx_j^2/\pi\bigr)\bigr)^{20} \quad (0 < x_j < \pi).$$

With $n = 100$, Ba and Joseph (2012) obtained *unnormalized* RMSE values of about 0.72–0.74 for CGP and about 0.72–0.88 for a GaSP(Const, Gauss) model over 50 repeat experiments.

This example demonstrates a virtue of using a normalized performance measure. To compute the normalizing factor for RMSE in (2.5), we followed the process of Ba and Joseph (2012). Training data from an LHD with $n = 100$ gives $\bar{y}$, the trivial predictor. The normalization in the denominator of (2.5) is computed from $N = 5000$ random test points. Repeating this process 50 times gives normalization factors of 0.71–0.74, about the same as the raw RMSE values from CGP. Thus, CGP's RMSE prediction accuracy is no better than that of the trivial $\bar{y}$ predictor, and the default method is worse. Effective prediction here is unattainable by CGP or GaSP and perhaps by no other approach with $n = 100$ because the function is so multi-modal; comparisons of CGP with other methods are meaningless in this example.

2-*d example*. For the function

$$y(x_1, x_2) = \sin\bigl(1/(x_1x_2)\bigr) \quad (0.3 \le x_j \le 1), \tag{5.1}$$

Ba and Joseph (2012) used a single design with $n = 24$ runs to compare CGP and GaSP(Const, Gauss). Their results suggest that accuracy is poor for both methods, which we confirmed. For this example, following Ba and Joseph (2012), a holdout set of 5000 random points on $[0.3, 1]^2$ was used. For one mLHD with $n = 24$, we obtained $e_{\mathrm{rmse,ho}}$ values of 0.23 and 0.24 for CGP and GaSP(Const, PowerExp), respectively. Moreover, the diagnostic plot in Figure 12 shows how badly CGP (and GaSP) perform. Both methods grossly overpredict for some points in the holdout set, with GaSP worse in this respect. Both methods also have large errors from under-prediction, with CGP worse.

Does this result generalize? With only two input variables and a function that is symmetric in x_1 and x_2, repeat experiments cannot be generated by permuting the column labels of the design. Reflecting within the x_1 and x_2 columns is considered below, but first we created multiple experiments by increasing n.

We were also curious about how large n has to be before acceptable accuracy is attained. Comparisons between CGP and GaSP(Const, PowerExp) were made for $n = 24, 25, \ldots, 48$; for each value of n an mLHD was generated. The $e_{\mathrm{rmse,ho}}$ results plotted in Figure 13(a) show that accuracy is not improved substantially for either method as n increases. Indeed,

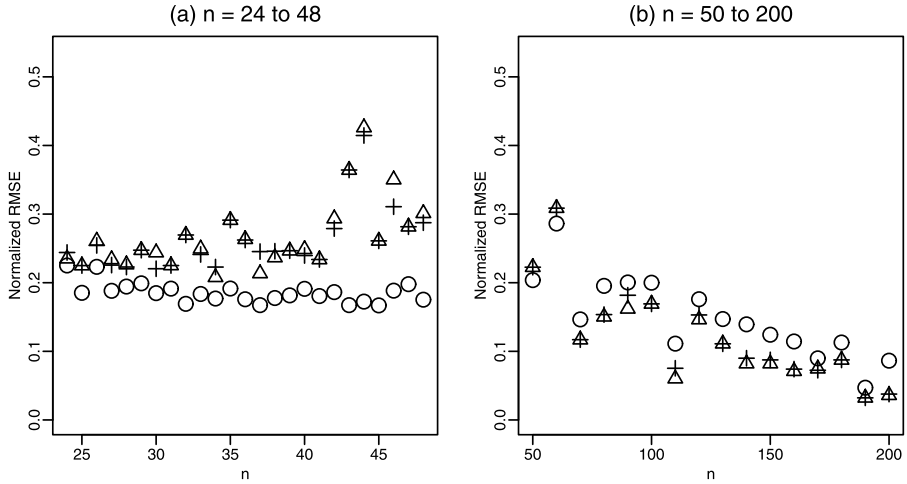

FIG. 13. *2-d function*: *Normalized holdout RMSE of prediction*, $e_{\text{rmse,ho}}$, *versus n for CGP* (○), *GaSP*(*Const*, *PowerExp*) (△), *and GaSP*(*Const*, *PowerExp*) *with nugget* (+).

GaSP(Const, PowerExp) gives variable accuracy, with larger values of n sometimes leading to worse accuracy than for $n = 24$. (The results in Figure 13 for a model with a nugget term are described in Section 5.3.)

To try to improve the accuracy, even larger sample sizes were tried. Figure 13(b) shows $e_{\text{rmse,ho}}$ for $n = 50, 60, \ldots, 200$. Both methods continue to give poor accuracy until n reaches 70, after which there is a slow, unsteady improvement. Curiously, GaSP now dominates.

Permuting between columns of a design does not generate distinct repeat experiments here, but reflecting either or both coordinates about the centers of their ranges maintains the distance properties of the design, that is, x_1 on [0.3, 1] is replaced by $x_1' = 1.3 - x_1$, and similarly x_2. Results for the repeat experiments from reflecting within x_1, x_2, or both x_1 and x_2 are available in the supplementary material (Chen et al., 2016). They are similar to those in Figure 13.

Thus, CGP dominates here for $n \leq 60$: it is inaccurate but less inaccurate than GaSP. For larger n, however, GaSP performs better, reaching the 0.10 threshold for $e_{\text{rmse,ho}}$ before CGP does. This example demonstrates the potential pitfalls of comparing two methods with a single experiment. A more comprehensive analysis not only gives more confidence in the findings but may also be essential to provide a balanced overview of advantages and disadvantages.

These last two toy functions together with the results in Figures 8–11 show no evidence for the effectiveness of a composite GaSP approach. These findings are in accord with the earlier study by West et al. (1995).

5.3 Adding a Nugget Term

A nugget augments the GaSP model in (2.1) with an uncorrelated ε term, usually assumed to have a normal distribution with mean zero and constant variance σ_ε^2, independent of the correlated process $Z(\mathbf{x})$. This changes the computation of $\mathbf{R}$ and $\mathbf{r}^T(\mathbf{x})$ in the conditional prediction (2.4), which no longer interpolates the training data. For data from physical experimentation or observation, augmenting a GaSP model in this way is natural to reflect random errors (e.g., Gao, Sacks and Welch, 1996; McMillan et al., 1999; Styer et al., 1995).

A nugget term has also been widely used for statistical modeling of deterministic computer codes without random error. The reasons offered are that numerical stability is improved, so overcoming computational obstacles, and also that a nugget can produce better predictive performance or better confidence or credibility intervals. The evidence—in the literature and presented here—suggests, however, that for deterministic functions the potential advantages of a nugget term are modest. More systematic methods are available to deal with numerical instability if it arises (Ranjan, Haynes and Karsten, 2011), adding a nugget does not convert a poor predictor into an acceptable one, and other factors may be more important for good statistical properties of intervals (Section 6.1). On the other hand, we also do not find that adding a nugget (and estimating it along with the other parameters) is harmful, though it may produce smoothers rather than interpolators. We now elaborate on these points.

A small nugget, that is, a small value of σ_ε^2, is often included to improve the numerical properties of $\mathbf{R}$. For

the space-filling initial designs in this article, however, Ranjan, Haynes and Karsten (2011) showed that ill-conditioning in a no-nugget GaSP model will only occur for low-dimensional $\mathbf{x}$, high correlation, and large n. These conditions are not commonly met in initial designs for applications. For instance, none of the computations for this article failed due to ill-conditioning, and those computations involved many repetitions of experiments for the various functions and GaSP models. The worst conditioning occurred for the 2-d example in Section 5.2 with $n = 200$, but even here the condition numbers of about 10^6 did not preclude reliable calculations. When a design is not space-filling, matrix ill-conditioning may indeed occur. For instance, a sequential design for, say, optimization or contour estimation (Bingham, Ranjan and Welch, 2014) could lead to runs close together in the $\mathbf{x}$ space, causing numerical problems. If ill-conditioning does occur, however, the mathematical solution proposed by Ranjan, Haynes and Karsten (2011) is an alternative to adding a nugget.

A nugget term is also sometimes suggested to improve predictive performance. Andrianakis and Challenor (2012) showed mathematically, however, that with a nugget the RMSE of prediction can be as large as that of a least squares fit of just the regression component in (2.1). Our empirical findings, choosing the size of σ_ε^2 via its MLE, are similarly unsupportive of a nugget. For example, the 2-d function in (5.1) is hard to predict with a GaSP(Const, PowerExp) model (Figure 13), but the results with a fitted nugget term shown by a "+" symbol in Figure 13 are no different in practice from those of the no-nugget model.

Similarly, repeating the calculations leading to Figure 2 for the borehole function, but fitting a nugget term in all models, shows essentially no difference [the results with a nugget are available in Figure 1 of the supplementary material (Chen et al., 2016)]. The MLE of σ_ε^2 is either zero or very small relative to the variance of the correlated process: typically $\hat{\sigma}_\varepsilon^2/\hat{\sigma}^2 < 10^{-6}$. These findings are consistent with those of Ranjan, Haynes and Karsten (2011), who found for the borehole function and other applications that constraining the model fit to have at least a modest value of σ_ε^2 deteriorated predictive performance.

Another example, the Friedman function,

$$y(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5, \tag{5.2}$$

with $n = 25$ runs, was used by Gramacy and Lee (2012) to illustrate potential advantages of including a nugget term. Their context—performance criteria, analysis method, and design—differs in all respects from ours. Our results in the top row of Figure 14 show that the GaSP(Const, Gauss) and GaSP(Const, PowerExp) models with $n = 25$ have highly variable accuracy, with $e_{\text{rmse,ho}}$ values no better and often much worse than 20%. The effect of the nugget is inconsequential. Increasing the sample size to $n = 50$ makes a dramatic improvement in prediction accuracy, but the effect of a nugget remains negligible.

The Gramacy and Lee (2012) results are not inconsistent with ours in that they did not report prediction accuracy for this example. Rather, their results relate to the role of the nugget in sometimes obtaining better uncertainty measures when a poor choice of correlation function is inadvertently made, a topic we return to in Section 6.1.

6. COMMENTS

6.1 Uncertainty of Prediction

As noted in Section 1, our attention is directed at prediction accuracy, the most compelling characteristic in practical settings. For example, where the objective is calibration and validation, the details of uncertainty, as distinct from accuracy, in the emulator of the computer model are absorbed (and usually swamped) by model uncertainties and measurement errors (Bayarri et al., 2007). But for specific predictions it is clearly important to have valid uncertainty statements.

Currently, a full assessment of the validity of emulator uncertainty quantification is unavailable. It has long been recognized that the standard error of prediction can be optimistic when MLEs of the parameters θ_j, p_j, ν_j in the correlation functions of Section 2.1 are "plugged-in" because the uncertainty in the parameter values is not taken into account (Abt, 1999). Corrections proposed by Abt remain to be done for the settings in which they are applicable.

Bayes credible intervals with full Bayes methods carry explicit and valid uncertainty statements; hybrid methods using priors on some of the correlation parameters (as distinct from MLEs) may also have reliable credible intervals. But for properties such as actual coverage probability (ACP), the proportion of points in a test set with true response values covered by intervals of nominal (say) 95% confidence or credibility, the behavior is far from clear. Chen (2013) compared several Bayes methods with respect to coverage. The results showed variability with respect to equivalent designs

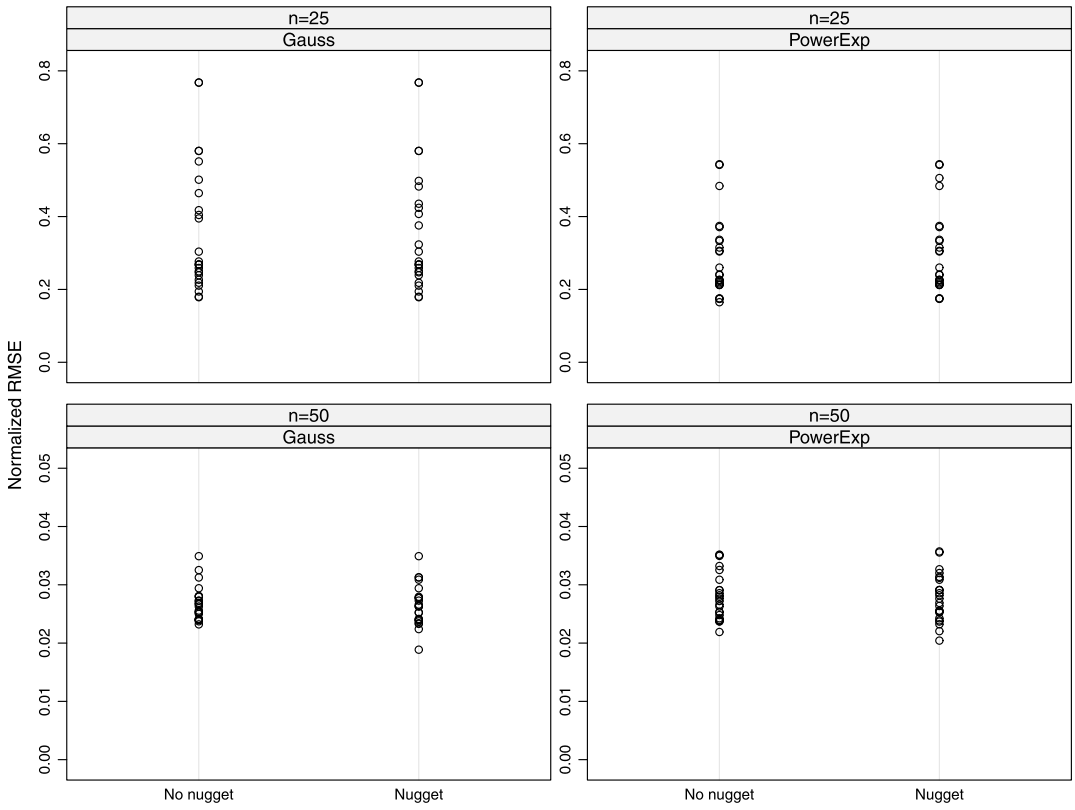

FIG. 14. *Friedman function*: *Normalized holdout RMSE of prediction*, $e_{\text{rmse,ho}}$, *for GaSP*(*Const*, *Gauss*) *and GaSP*(*Const*, *PowerExp*) *models with no nugget term versus the same models with a nugget. There are two base designs*: *a* 25*-run mLHD* (*top row*); *and a* 50*-run mLHD* (*bottom row*). *For each base design*, 25 *random permutations between columns give the* 25 *values of* $e_{\text{rmse,ho}}$ *in a dot plot*.

like that found above for accuracy, a troubling characteristic pointing to considerable uncertainty about the uncertainty.

In Figure 15 we see some of the issues. It gives ACP results for the borehole and Nilson–Kuusk functions. The left-hand plot for borehole displays the anticipated under-coverage using plug-in estimates for the correlation parameters. (Confidence intervals here use $n-1$ rather than n in the estimate of σ in the standard error and t_{n-1} instead of the standard normal.) PowerExp is slightly better than Gauss, and Bayes-GEM-SA has ACP values close to the nominal 95%. Surprisingly, the plot for the Nilson–Kuusk code on the right of Figure 15 paints a different picture. Plug-in with Gauss and Bayes-GEM-SA both show under-coverage, while plug-in PowerExp has near-ideal properties here. We speculate that the use of the Gauss correlation function by Bayes-GEM-SA is again suboptimal for the Nilson–Kuusk application, just as it was for prediction accuracy.

The supplementary material (Chen et al., 2016) compares models with and without a nugget in terms of coverage properties for the Friedman function in (5.2). The results show that the problem of substantial under-coverage seen in many of the replicate experiments is not solved by inclusion of a nugget term. A modest improvement in the distribution of ACP values is seen, particularly for $n = 50$, an improvement consistent with the advantage seen in Table 1 of Gramacy and Lee (2012) from fitting a nugget term.

A more complete study is surely needed to clarify appropriate criteria for uncertainty assessment and how modeling choices may affect matters.

6.2 Extrapolation

GaSP based methods are interpolations so our findings are clearly limited to prediction in the space of the experiment. The design of the computer experiment should cover the region of interest, rendering extrapolation meaningless. If a new region of interest is found, for example, during optimization, the initial computer runs can be augmented; extrapolation can be used to delimit regions that have to be explored further. Of course, extrapolation is necessary in the situation of a

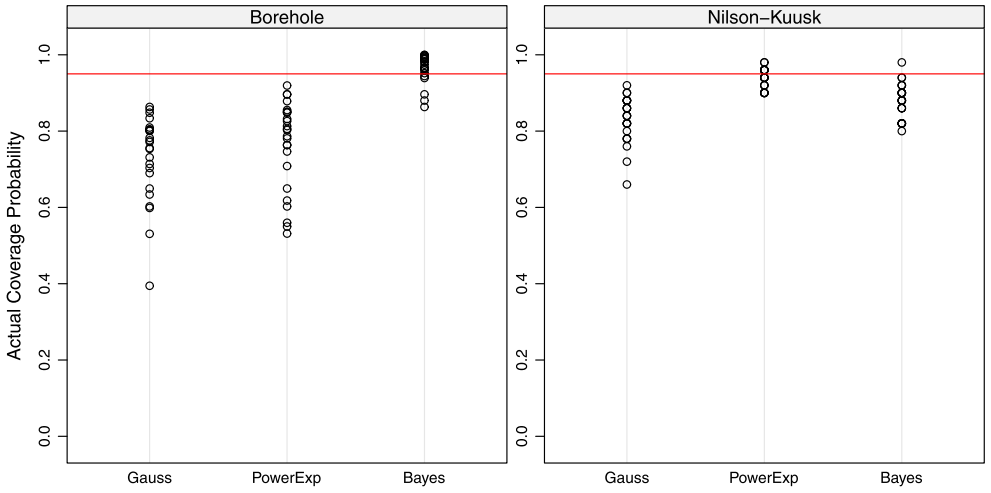

FIG. 15. *Borehole and Nilson–Kuusk functions*: *ACP of nominal* 95% *confidence or credibility intervals for GaSP*(*Const*, *Gauss*), *GaSP*(*Const*, *PowerExp*), *and Bayes-GEM-SA. For the borehole function*, 25 *random permutations between columns of a* 40*-run mLHD give the* 25 *values of ACP in a dot plot. For the Nilson–Kuusk function*, 25 *designs are created from a* 150*-run LHD base plus* 50 *random points from a* 100*-run LHD. The remaining* 50 *points in the* 100*-run LHD form the holdout set for each repeat.*

new region and a code that can no longer be run. But then the question is how to extrapolate. Initial inclusion of linear or other regression terms may be more useful than just a constant, but it may also be useless, or even dangerous, unless the "right" extrapolation terms are identified. We suspect it would be wiser to examine main effects resulting from the application of GaSP and use them to guide extrapolation.

6.3 Performance Criteria

We have focused almost entirely on questions of predictive accuracy and used RMSE as a measure. The supplementary material (Chen et al., 2016) defines and provides results for a normalized version of maximum absolute error, $e_{\max,\mathrm{ho}}$. Other computations we have done use the median of the absolute value of prediction errors, with normalization relative to the trivial predictor from the median of the training output data. These results are qualitatively the same as for $e_{\mathrm{rmse,ho}}$: regression terms do not matter, and PowerExp is a reliable choice for R. For slow codes, analysis like in Section 4 but using $e_{\max,\mathrm{ho}}$ has some limited value in identifying regions where predictions are difficult, the limitations stemming from a likely lack of coverage of subregions, especially at borders of the unit cube, where the output function may behave badly.

A common performance measure for slow codes uses leave-one-out cross-validation error to produce analogues of $e_{\mathrm{rmse,ho}}$ and $e_{\max,\mathrm{ho}}$, obviating the need for a holdout set. For fast codes, repeat experiments and the ready availability of a holdout set render cross-validation unnecessary, however. For slow codes with only one set of data available, the single assessment from leave-one-out cross-validation does not reflect the variability caused, for example, by the design. In any case, qualitatively similar conclusions pertain regarding regression terms and correlation functions.

6.4 More Examples

The examples we selected are codes that have been used in earlier studies. We have not incorporated 1-d examples; while instructive for pedagogical reasons, they have little presence in practice. Other applications we could have included (e.g., Gough and Welch, 1994) duplicate the specific conclusions we draw below. There are also "fabricated" test functions in the numerical integration and interpolation literature (Barthelmann, Novak and Ritter, 2000) and some specifically for computer experiments (Surjanovic and Bingham, 2015). They exhibit characteristics sometimes similar to those in Section 5—large variability in a corner of the space, a condition that inhibits and even prevents construction of effective surrogates—and sometimes no different than the examples in Section 3. Codes that are deterministic but with numerical errors could also be part of a diverse catalogue of test problems. Ideally performance metrics from various approaches would be provided to facilitate comparisons; the suite of examples that we employed is a starting point.

6.5 Designs

The variability in performance over equivalent designs is a striking phenomenon in the analyses of Section 3 and raises questions about how to cope with what seems to be unavoidable bad luck. Are there sequential strategies that can reduce the variability? Are there advantageous design types, more robust to arbitrary symmetries. For example, does it matter whether a random LHD, mLHD, or an orthogonal LHD is used? The latter question is currently being explored by the authors. That design has a strong role is both unsurprising and surprising. It is not surprising that care must be taken in planning an experiment; it is surprising and perplexing that equivalent designs can lead to such large differences in performance that are not mediated by good analytic procedures.

6.6 Larger Sample Sizes

As noted in Section 1, our attention is on experiments where n is small or modest at most. With advances in computing power it becomes more feasible to mount experiments with larger values of n while, at the same time, more complex codes become feasible but only with limited n. Our focus continues to be on the latter and the utility of GaSP models in that context.

As n gets larger, Figure 2 illustrates that the differences in accuracy among choices of R and μ begin to vanish. Indeed, it is not even clear that using GaSP models for large n is useful; standard function fitting methods such as splines may well be competitive and easier to compute. In addition, when n is large nonstationary behavior can become apparent and encourages variations in the GaSP methodology such as decomposing the input space (as in Gramacy and Lee, 2008) or by using a complex μ together with a computationally more tractable R (as in Kaufman et al., 2011). Comparison of alternatives when n is large is yet to be considered.

6.7 Are Regression Terms Ever Useful?

Introducing regression terms is unnecessary in the examples we have presented; a heuristic rationale was given in Section 2.2. The supplementary material (Chen et al., 2016) reports a simulation study with realized functions generated as follows: (1) there are very large linear trends for all x_j; and (2) the superimposed sample path from a 0-mean GP is highly nonlinear, that is, a GP with at least one $\theta_j \gg 0$ in (2.2). Even under such extreme conditions, the advantage of explicitly fitting the regression terms is limited to a relative (ratio of $e_{\text{rmse,ho}}$) advantage, with only small differences in $e_{\text{rmse,ho}}$; the presence of a large trend causes a large normalizing factor. Moreover, such functions are not the sort usually encountered in computer experiments. If they do show up, standard diagnostics will reveal their presence and allow effective follow-up analysis (see Section 7.2).

7. CONCLUSIONS AND RECOMMENDATIONS

This article addresses two types of questions. First, how should the analysis methodologies advanced in the study of computer experiments be assessed? Second, what recommendations for modeling strategies follow from applying the assessment strategy to the particular codes we have studied?

7.1 Assessing Methods

We have stressed the importance of going beyond "anecdotes" in making claims for proposed methods. While this point is neither novel nor startling, it is one that is commonly ignored, often because the process of studying consequences under multiple conditions is more laborious. The borehole example (Figure 2), for instance, employs 75 experiments arising from 25 repeats of each of 3 base experiments.

When only one training set of data is available (as can be the case with slow codes), the procedures in Section 4, admittedly ad hoc, nevertheless expand the range of conditions. This brings more generalizability to claims about the comparative performances of competing procedures. The same strategy of creating multiple training/holdout sets is potentially useful in comparing competing methods in physical experiments as well.

The studies in the previous sections lead to the following conclusions:

- There is no evidence that GaSP(Const, PowerExp) is ever dominated by use of regression terms, or other choices of R. Moreover, we have found that the inclusion of regression terms makes the likelihood surface multi-modal, necessitating an increase in computational effort for maximum likelihood or Bayesian methods. This appears to be due to confounding between regression terms and the GP paths.
- Choosing $R =$ Gauss, though common, can be unwise. The Matérn function optimized over a few levels of smoothness is a reasonable alternative to PowerExp.
- Design matters but cannot be controlled completely. Variability of performance from equivalent designs can be uncomfortably large.

There is not enough evidence to settle the following questions:

- Are full Bayes methods ever more accurate than GaSP(Const, PowerExp)? Bayes methods relying on $R =$ Gauss were seen to be sometimes inferior, and extensions to accommodate less smooth R such as PowerExp, perhaps via hybrid Bayes-MLE methods, are needed.
- Are composite GaSP methods ever better than GaSP(Const, PowerExp) in practical settings where the output exhibits nonstationary behavior?

7.2 Recommendations

Faced with a particular code and a set of runs, what should a scientist do to produce a good predictor? Our recommendation is to make use of GaSP(Const, PowerExp), use the diagnostics of Jones, Schonlau and Welch (1998) or Bastos and O'Hagan (2009), and assess whether the GaSP predictor is adequate. If found inadequate, then the scientist should expect no help from introducing regression terms and, until further evidence is found, neither from Bayes nor CGP approaches. Of course, trying such methods is not prohibited, but we believe that inadequacy of the GaSP(Const, PowerExp) model is usually a sign that more substantial action must be taken.

We conjecture that the best way to proceed in the face of inadequacy is to devise a second (or multiple) stage process, perhaps by added runs, or perhaps by carving the space into more manageable subregions as well as adding runs. How best to do this has been partially addressed, for example, by Gramacy and Lee (2008) and Loeppky, Moore and Williams (2010); effective methods constrained by limited runs are not apparent and in need of study.

ACKNOWLEDGMENTS

We thank the referees, Associate Editor, and Editor for suggestions that clarified and broadened the scope of the studies reported here.

The research of Loeppky and Welch was supported in part by grants from the Natural Sciences and Engineering Research Council, Canada.

SUPPLEMENTARY MATERIAL

Supplement to "Analysis Methods for Computer Experiments: How to Assess and What Counts?" (DOI: 10.1214/15-STS531SUPP; .zip). This report (whatcounts-supp.pdf) contains further description of the test functions and data from running them, further results for root mean squared error, findings for maximum absolute error, further results on uncertainty of prediction, and details of the simulation investigating regression terms. Inputs to the Arctic sea-ice code—ice-x.txt. Outputs from the code—ice-y.txt.

REFERENCES

ABT, M. (1999). Estimating the prediction mean squared error in Gaussian stochastic processes with exponential correlation structure. *Scand. J. Stat.* **26** 563–578. MR1734262

ANDRIANAKIS, I. and CHALLENOR, P. G. (2012). The effect of the nugget on Gaussian process emulators of computer models. *Comput. Statist. Data Anal.* **56** 4215–4228. MR2957866

BA, S. and JOSEPH, V. R. (2012). Composite Gaussian process models for emulating expensive functions. *Ann. Appl. Stat.* **6** 1838–1860. MR3058685

BARTHELMANN, V., NOVAK, E. and RITTER, K. (2000). High dimensional polynomial interpolation on sparse grids. *Adv. Comput. Math.* **12** 273–288. MR1768951

BASTOS, L. S. and O'HAGAN, A. (2009). Diagnostics for Gaussian process emulators. *Technometrics* **51** 425–438. MR2756478

BAYARRI, M. J., BERGER, J. O., PAULO, R., SACKS, J., CAFEO, J. A., CAVENDISH, J., LIN, C.-H. and TU, J. (2007). A framework for validation of computer models. *Technometrics* **49** 138–154. MR2380530

BAYARRI, M. J., BERGER, J. O., CALDER, E. S., DALBEY, K., LUNAGOMEZ, S., PATRA, A. K., PITMAN, E. B., SPILLER, E. T. and WOLPERT, R. L. (2009). Using statistical and computer models to quantify volcanic hazards. *Technometrics* **51** 402–413. MR2756476

BINGHAM, D., RANJAN, P. and WELCH, W. J. (2014). Design of computer experiments for optimization, estimation of function contours, and related objectives. In *Statistics in Action* (J. F. Lawless, ed.) 109–124. CRC Press, Boca Raton, FL. MR3241971

CHAPMAN, W. L., WELCH, W. J., BOWMAN, K. P., SACKS, J. and WALSH, J. E. (1994). Arctic sea ice variability: Model sensitivities and a multidecadal simulation. *J. Geophys. Res.* **99C** 919–935.

CHEN, H. (2013). Bayesian prediction and inference in analysis of computer experiments. Master's thesis, Univ. British, Columbia, Vancouver.

CHEN, H., LOEPPKY, J. L., SACKS, J. and WELCH, W. J. (2016). Supplement to "Analysis Methods for Computer Experiments: How to Assess and What Counts?" DOI:10.1214/15-STS531SUPP.

CURRIN, C., MITCHELL, T., MORRIS, M. and YLVISAKER, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *J. Amer. Statist. Assoc.* **86** 953–963. MR1146343

DIXON, L. C. W. and SZEGÖ, G. P. (1978). The global optimisation problem: An introduction. In *Towards Global Optimisation* (L. C. W. Dixon and G. P. Szegö, eds.) 1–15. North Holland, Amsterdam.

GAO, F., SACKS, J. and WELCH, W. J. (1996). Predicting urban ozone levels and trends with semiparametric modeling. *J. Agric. Biol. Environ. Stat.* **1** 404–425. MR1807773

GOUGH, W. A. and WELCH, W. J. (1994). Parameter space exploration of an ocean general circulation model using an isopycnal mixing parameterization. *J. Mar. Res.* **52** 773–796.

GRAMACY, R. B. and LEE, H. K. H. (2008). Bayesian treed Gaussian process models with an application to computer modeling. *J. Amer. Statist. Assoc.* **103** 1119–1130. MR2528830

GRAMACY, R. B. and LEE, H. K. H. (2012). Cases for the nugget in modeling computer experiments. *Stat. Comput.* **22** 713–722. MR2909617

JONES, D. R., SCHONLAU, M. and WELCH, W. J. (1998). Efficient global optimization of expensive black-box functions. *J. Global Optim.* **13** 455–492. MR1673460

JOSEPH, V. R., HUNG, Y. and SUDJIANTO, A. (2008). Blind kriging: A new method for developing metamodels. *J. Mech. Des.* **130** 031102–1–8.

KAUFMAN, C. G., BINGHAM, D., HABIB, S., HEITMANN, K. and FRIEMAN, J. A. (2011). Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology. *Ann. Appl. Stat.* **5** 2470–2492. MR2907123

KENNEDY, M. (2004). Description of the Gaussian process model used in GEM-SA. Techical report, Univ. Sheffield. Available at http://www.tonyohagan.co.uk/academic/GEM/.

LIM, Y. B., SACKS, J., STUDDEN, W. J. and WELCH, W. J. (2002). Design and analysis of computer experiments when the output is highly correlated over the input space. *Canad. J. Statist.* **30** 109–126. MR1907680

LOEPPKY, J. L., MOORE, L. M. and WILLIAMS, B. J. (2010). Batch sequential designs for computer experiments. *J. Statist. Plann. Inference* **140** 1452–1464. MR2592224

LOEPPKY, J. L., SACKS, J. and WELCH, W. J. (2009). Choosing the sample size of a computer experiment: A practical guide. *Technometrics* **51** 366–376. MR2756473

MCMILLAN, N. J., SACKS, J., WELCH, W. J. and GAO, F. (1999). Analysis of protein activity data by Gaussian stochastic process models. *J. Biopharm. Statist.* **9** 145–160.

MORRIS, M. D. and MITCHELL, T. J. (1995). Exploratory designs for computational experiments. *J. Statist. Plann. Inference* **43** 381–402.

MORRIS, M. D., MITCHELL, T. J. and YLVISAKER, D. (1993). Bayesian design and analysis of computer experiments: Use of derivatives in surface prediction. *Technometrics* **35** 243–255. MR1234641

NILSON, T. and KUUSK, A. (1989). A reflectance model for the homogeneous plant canopy and its inversion. *Remote Sens. Environ.* **27** 157–167.

O'HAGAN, A. (1992). Some Bayesian numerical analysis. In *Bayesian Statistics*, 4 (*PeñíScola*, 1991) (J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith, eds.) 345–363. Oxford Univ. Press, New York. MR1380285

PICHENY, V., GINSBOURGER, D., RICHET, Y. and CAPLIN, G. (2013). Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics* **55** 2–13. MR3038476

PRESTON, D. L., TONKS, D. L. and WALLACE, D. C. (2003). Model of plastic deformation for extreme loading conditions. *J. Appl. Phys.* **93** 211–220.

RANJAN, P., HAYNES, R. and KARSTEN, R. (2011). A computationally stable approach to Gaussian process interpolation of deterministic computer simulation data. *Technometrics* **53** 366–378. MR2850469

SACKS, J., SCHILLER, S. B. and WELCH, W. J. (1989). Designs for computer experiments. *Technometrics* **31** 41–47. MR0997669

SACKS, J., WELCH, W. J., MITCHELL, T. J. and WYNN, H. P. (1989). Design and analysis of computer experiments (with discussion). *Statist. Sci.* **4** 409–435. MR1041765

SCHONLAU, M. and WELCH, W. J. (2006). Screening the input variables to a computer model via analysis of variance and visualization. In *Screening*: *Methods for Experimentation in Industry*, *Drug Discovery*, *and Genetics* (A. Dean and S. Lewis, eds.) 308–327. Springer, New York.

STEIN, M. L. (1999). *Interpolation of Spatial Data*: *Some Theory for Kriging*. Springer, New York. MR1697409

STYER, P., MCMILLAN, N., GAO, F., DAVIS, J. and SACKS, J. (1995). Effect of outdoor airborne particulate matter on daily death counts. *Environ. Health Perspect.* **103** 490–497.

SURJANOVIC, S. and BINGHAM, D. (2015). Virtual library of simulation experiments: Test functions and datasets. Available at http://www.sfu.ca/~ssurjano.

WELCH, W. J., BUCK, R. J., SACKS, J., WYNN, H. P., MITCHELL, T. J. and MORRIS, M. D. (1992). Screening, predicting, and computer experiments. *Technometrics* **34** 15–25.

WELCH, W. J., BUCK, R. J., SACKS, J., WYNN, H. P., MORRIS, M. D. and SCHONLAU, M. (1996). Response to James M. Lucas. *Technometrics* **38** 199–203.

WEST, O. R., SIEGRIST, R. L., MITCHELL, T. J. and JENKINS, R. A. (1995). Measurement error and spatial variability effects on characterization of volatile organics in the subsurface. *Environ. Sci. Technol.* **29** 647–656.